# SE 2A04 Fall 2000

## Lab Exercise 3

Instructor: William M. Farmer

Assigned: 10 October 2000
Lab report due: 24 October 2000

### Step 1

Below is an interface for a module of tree functions. The definitions from Lab Exercise 2 are assumed. With your partner, produce two different implementations of the module. The two implementations should use different internal representations of trees.

**Interface:**

1. store : strings $\rightarrow$ **Z** is a unary function that, given a string representing a tree (according to the scheme of Lab Exercise 2), stores the tree and then returns 1 and, given a string that does not represent any tree, returns -1.

2. check : $\rightarrow$ **Z** is a nullary function that returns 1 if a tree that is a pair is stored; 0 if a tree that is an integer is stored; and -1 if no tree is stored.

3. left : $\rightarrow$ strings is a nullary function that returns a string representing the left component of the stored tree.

4. right : $\rightarrow$ strings is a nullary function that returns a string representing the right component of the stored tree.

5. height : $\rightarrow$ **Z** is a nullary function that returns the height of the stored tree.

6. size : $\rightarrow$ **Z** is a nullary function that returns the number of pairs in the stored tree.

### Step 2

As a separate program, write a user interface program that allows users to store a tree and to inquire about the characteristics of the tree. The user interface program must accomplish its task by using the module described above. It should inform the user in a friendly way of any errors in input.

### Step 3

Test both implementations of the module and the user interface program separately until you are satisfied that they are correct. Then combine them to produce two full programs that can be demonstrated. Test the two full programs.

### Step 4

During the lab session on 17 October 2000, exchange your two module implementations with those of another team chosen by the instructor or TAs. Test them separately in the same way as in Step 3. Then combine your user interface program with each of the module implementations to produce two full programs. Test the full programs in the same way as in Step 3.

### Step 5

Write a lab report that includes the following:

1. A description and analysis of your work and the results obtained. In particular, analyze the difficulties you encountered and explain how they could have been avoided.

2. Listings of your source code and documentation. Be prepared to provide copies of your source code files if they are requested. Your two full programs must work correctly on the Sun workstations in the laboratory.

3. A description of how you tested your programs including what test data you used.

4. A copy of each team member's log book.

The final lab report is due at the beginning of the lab session on 24 October 2000.

### Step 6

During the lab session on 24 October 2000, each team will be required to demonstrate their two full programs (each with a different implementation of the tree module) to one of the TAs.